

Hardness of RNA Folding Problem with Four Symbols

Yi-Jun Chang*

University of Michigan
cyijun@umich.edu

Abstract. An RNA sequence is a string composed of four types of nucleotides, A, C, G , and U . Given an RNA sequence, the goal of the RNA folding problem is to find a maximum cardinality set of crossing-free pairs of the form $\{A, U\}$ or $\{C, G\}$. The problem is central in bioinformatics and has received much attention over the years. However, the current best algorithm for the problem still takes $\mathcal{O}\left(\frac{n^3}{\log^2(n)}\right)$ time, which is only a slight improvement over the classic $\mathcal{O}(n^3)$ dynamic programming algorithm. Whether the RNA folding problem can be solved in $\mathcal{O}(n^{3-\epsilon})$ time remains an open problem. Recently, Abboud, Backurs, and Williams (FOCS'15) made the first progress by showing a conditional lower bound for a generalized version of the RNA folding problem based on a conjectured hardness of the k -clique problem. A drawback of their work is that they require the RNA sequence to have at least 36 types of letters, making their result biologically irrelevant. In this paper, we show that by constructing the gadgets using a lemma of Bringmann and Künnemann (FOCS'15) and surrounding them with some carefully designed sequences, the framework of Abboud et al. can be improved upon to work for the case where the alphabet size is 4, yielding a conditional lower bound for the RNA folding problem. We also investigate the Dyck edit distance problem. We demonstrate a reduction from RNA folding problem to Dyck edit distance problem of alphabet size 10, establishing a connection between the two fundamental string problems. This leads to a much simpler proof of the conditional lower bound for Dyck edit distance problem given by Abboud et al. and lowers the required alphabet size for the lower bound to work.

Keywords: RNA folding, Dyck edit distance, longest common subsequence, conditional lower bound, clique

1. Introduction

An *RNA sequence* is a string composed of four types of nucleotides, namely A, C, G , and U . Given an RNA sequence, the goal of the *RNA folding* problem is to find a maximum cardinality set of crossing-free pairs of nucleotides, where all the pairs are either $\{A, U\}$ or $\{C, G\}$. The problem is central in bioinformatics and has found applications in many areas of molecular biology. For a more comprehensive exposition of the topic, the reader is referred to e.g. [20].

It is well-known that the problem can be solved in cubic time using a simple dynamic programming method [10]. Due to the importance of RNA folding in practice, there has been a long line of research on improving the cubic time algorithm (See e.g. [1,12,15,16,20,21]). Currently the best upper bound is $\mathcal{O}\left(\frac{n^3}{\log^2(n)}\right)$ [16,20], and this can be obtained via four-Russian method or fast min-plus multiplication (based on ideas from Valiant's CFG parser [22]).

Whether the RNA folding problem can be solved in $\mathcal{O}(n^{3-\epsilon})$ time for some $\epsilon > 0$ is still a major open problem. Other than attempting to improve the upper bound, we should also approach the problem in the opposite direction, i.e. showing a lower bound or arguing why the problem is hard.

A popular way to show hardness of a problem is to demonstrate a lower bound conditioned on some widely accepted hypothesis.

* Supported by NSF grants CCF-1217338, CNS-1318294, and CCF-1514383.

Conjecture 1 (Strongly Exponential Time Hypothesis (SETH)). There exists no $\epsilon, k_0 > 0$ such that k -SAT with n variables can be solved in time $\mathcal{O}(2^{(1-\epsilon)n})$ for all $k > k_0$.

Conjecture 2. There exists no $\epsilon, k_0 > 0$ such that k -clique on graphs with n nodes can be solved in time $\tilde{\mathcal{O}}(n^{(\omega-\epsilon)k/3})$ for all $k > k_0$, where $\omega < 2.373$ is the matrix multiplication exponent.

Assuming that SETH (Conjecture 1) holds, the following bounds are unattainable for any $\epsilon > 0$:

- an $\mathcal{O}(n^{k-\epsilon})$ algorithm for k -dominating set problem [14],
- an $\mathcal{O}(n^{2-\epsilon})$ algorithm for dynamic time warping, longest common subsequence, and edit distance [4,7,8],
- an $\mathcal{O}(m^{2-\epsilon})$ algorithm for $(3/2 - \epsilon)$ -approximating the diameter of a graph with m edges [17].

As remarked in [3], it is easy to reduce the longest common subsequence problem on binary strings to the RNA folding problem as following: Given two binary strings X, Y , we let $\hat{X} \in \{A, C\}^{|X|}$ be the string such that $\hat{X}[i] = A$ if $X[i] = 0$, $\hat{X}[i] = C$ if $X[i] = 1$, and we let $\hat{Y} \in \{G, U\}^{|Y|}$ be the string such that $\hat{Y}[i] = U$ if $Y[i] = 0$, $\hat{Y}[i] = G$ if $Y[i] = 1$. Then we have a 1-1 correspondence between RNA foldings of $\hat{X} \circ \hat{Y}^R$ (i.e. concatenation of \hat{X} and the reversal of \hat{Y}) and common subsequences of X and Y . It has been shown in [8] that there is no $\mathcal{O}(n^{2-\epsilon})$ algorithm for longest common subsequence problem on binary strings conditioned on SETH, and we immediately get the same conditional lower bound for RNA folding from the simple reduction!

Very recently, based on a conjectured hardness of k -clique problem (Conjecture 2), a higher conditional lower bound was proved for a generalized version of the RNA folding problem (which coincides with the RNA folding problem when the alphabet size is 4) [3]:

Theorem 1 ([3]). *If the generalized RNA folding problem on sequences of length n with alphabet size 36 can be solved in $T(n)$ time, then $3k$ -clique on graphs with $|V| = n$ can be solved in $\mathcal{O}(T(n^{k+2} \log(n)))$ time.*

Therefore, a $\mathcal{O}(n^{\omega-\epsilon})$ time algorithm for the generalized RNA folding with alphabet size at least 36 will disprove Conjecture 2, yielding a breakthrough to the parameterized complexity of clique problem.

However, the above theorem is irrelevant to the RNA folding problem in real life (which has alphabet size 4). It is unknown whether the generalized RNA folding for alphabet size 4 admits a faster algorithm than the case for alphabet size > 4 . In fact, there are examples of string algorithms whose running time scales with alphabet size (e.g. string matching with mismatched [6] and jumbled indexing [5,9]). We also note that when the alphabet size is 2, the generalized RNA folding can be trivially solved in linear time.

In this paper, we improve upon Theorem 1 by showing the same conditional lower bound for the RNA folding problem:

Theorem 2. *If the RNA folding problem on sequences in $\{A, C, G, U\}^n$ can be solved in $T(n)$ time, then $3k$ -clique on graphs with $|V| = n$ can be solved in $\mathcal{O}(T(n^{k+1} \log(n)))$ time.*

Note that we also get an $\mathcal{O}(n)$ factor improvement inside $T(\cdot)$, though it does not affect the conditional lower bound.

The current state-of-art algorithm for k -clique, which takes $\tilde{\mathcal{O}}(n^{\omega k/3})$ time, requires the use of fast matrix multiplication [11] which does not perform very efficiently in practice. For combinatorial, non-algebraic algorithm for k -clique, the current best one runs in $\tilde{\mathcal{O}}\left(\frac{n^k}{\log^{k(n)}}\right)$ time [23], which is only slightly better than the trivial approach. As a result, by Theorem 2, even a $\mathcal{O}(n^{3-\epsilon})$ time

combinatorial algorithm for RNA folding will lead to an improvement for combinatorial algorithms for k -clique!

In the proof of Theorem 1 in [3], given a graph $G = (V, E)$, a sequence of length $\mathcal{O}(n^{k+2} \log(n))$ is constructed in such a way that we can decide whether G has a $3k$ -clique according to the number of pairs in an optimal generalized RNA folding of S . Such a construction requires many different types of letters in order to build various “walls” which prevent undesired pairings between different parts of the sequence. Hence extending their approach to handle the case where the alphabet size is 4 may not be easy without aid from other techniques and ideas.

Overview of our approach. At a high level, our reduction (from $3k$ -clique problem to RNA folding problem) follows the approach in [3]: We enumerate all k -cliques, and each of them is encoded as some gadgets. All the gadgets are then put together to form an RNA sequence. The goal is to ensure that an optimal RNA folding corresponds to choosing three k -cliques that form a $3k$ -clique, given that the underlying graph admits a $3k$ -clique.

To achieve this goal without using extra types of letters that force the gadgets to match in a desired manner, we construct the gadgets via a key lemma in [8], whose original purpose is to prove that longest common subsequence and other edit distance problems are SETH-hard even on binary strings. We will treat it as a black box and apply it multiple times during the construction. This powerful tool will allow us to test whether two k -cliques form a $2k$ -clique by the longest common subsequence between the two strings representing the two k -cliques.

In the final RNA sequence, all clique gadgets are well-separated by some carefully designed sequences whose purpose is to “trap” all the clique gadgets except three of them. Since we know that these three clique gadgets are guaranteed to match well if the graph has a $3k$ -clique, we can infer whether the graph has a $3k$ -clique from the optimal RNA folding of the RNA sequence.

Dyck Edit Distance. One other way to formulate the RNA folding problem is as follows: deleting the minimum number of letters in a given string to transform the string into a string in the language defined by the grammar $\mathbf{S} \rightarrow \mathbf{SS}, \mathbf{ASU}, \mathbf{USA}, \mathbf{CSG}, \mathbf{GSC}, \epsilon$ (empty string). The *Dyck edit distance problem* [18,19], which asks for the minimum number of edits to transform a given string to a well-balanced parentheses of s different types, has a similar formulation. Due to the similarity, the same conditional lower bound as Theorem 1 was also shown for the Dyck edit distance problem (with alphabet size ≥ 48) in [3].

In this paper, we improve and simplify their result by demonstrating a simple reduction from RNA folding to Dyck edit distance problem:

Theorem 3. *If Dyck edit distance problem on sequences of length n with alphabet size 10 can be solved in $T(n)$ time, then the RNA folding problem on sequences in $\{A, C, G, U\}^n$ can be solved in $\mathcal{O}(T(n))$ time.*

Combining Theorem 2, 3, we get the following corollary:

Corollary 1. *If the Dyck edit distance problem on sequences of length n with alphabet size 10 can be solved in $T(n)$ time, then $3k$ -clique on graphs with $|V| = n$ can be solved in $\mathcal{O}(T(n^{k+1} \log(n)))$ time.*

2. Preliminaries

Given a set of letters Σ , the set Σ' is defined as $\{x' | x \in \Sigma\}$. We require that $\Sigma \cap \Sigma' = \emptyset$, and $\forall x, y \in \Sigma, (x \neq y) \rightarrow (x' \neq y')$. Therefore, we have $|\Sigma'| = |\Sigma|$ and $|\Sigma \cup \Sigma'| = 2|\Sigma|$.

For any $X = (x_1, \dots, x_k) \in \Sigma^k$, we write $p(X)$ to denote (x'_1, \dots, x'_k) (the letter p stands for the prime symbol). We denote the reversal of the sequence X as X^R . The concatenation of two sequences X, Y is denoted as $X \circ Y$ (or simply XY). We write *substring* to denote a contiguous subsequence.

Two pairs of indices $(i_1, j_1), (i_2, j_2)$, with $i_1 < j_1$ and $i_2 < j_2$, form a *crossing pair* iff

$$(\{i_1, j_1\} \cap \{i_2, j_2\} \neq \emptyset) \vee (i_1 < i_2 < j_1 < j_2) \vee (i_2 < i_1 < j_2 < j_1).$$

Generalized RNA Folding. Given $S \in (\Sigma \cup \Sigma')^n$, the goal of the generalized RNA folding problem is to find a maximum cardinality set $A \subseteq \{(i, j) | 1 \leq i < j \leq n\}$ among all sets meeting the following conditions:

- A does not contain any crossing pair.
- For any $(i, j) \in A$, either (i) $S[i] \in \Sigma$ and $S[j] = S[i]'$ or (ii) $S[j] \in \Sigma$ and $S[i] = S[j]'$ is true.

We write $\text{RNA}(S) = |A|$.

Any set meeting the above conditions is called an *RNA folding* of S . If its cardinality equals $\text{RNA}(S)$, then it is said to be *optimal*.

In the paper we will only focus on the generalized RNA folding problem with four types of letters, i.e. $\Sigma = \{0, 1\}, \Sigma' = \{0', 1'\}$, which coincides with the RNA folding problem for alphabet $\{A, C, G, U\}$.

With a slight abuse of notation, sometimes we will write $(S[i], S[j])$ to denote a pair $(i, j) \in A$. The notation $\{\cdot, \cdot\}$ is used to indicate an unordered pair.

Longest Common Subsequence (LCS). Given $X \in \Sigma^n$ and $Y \in \Sigma^m$, we define $\delta_{\text{LCS}}(X, Y) = n + m - 2k$, where k = the length of the longest common subsequence of X and Y . It is easy to observe that $\text{RNA}(X \circ p(Y^R))$ equals the length of $\text{LCS} = (n + m - \delta_{\text{LCS}}(X, Y))/2$. In this sense, we can conceive of an LCS problem as an RNA folding problem with some structural constraint on the sequence.

In [8], a conditional lower bound for the LCS problem with $|\Sigma| = 2$ based on SETH was presented. A key technique in their approach is a function that transforms an instance of an alignment problem between two sets of sequences to an instance of the LCS problem.

Alignments of two sets of sequences. Let $\mathbf{X} = (X_1, \dots, X_n)$ and $\mathbf{Y} = (Y_1, \dots, Y_m)$ be two linearly ordered sets of sequences of alphabet Σ . We assume that $n \geq m$. An *alignment* is a set $A = \{(i_1, j_1), (i_2, j_2), \dots, (i_{|A|}, j_{|A|})\}$ with $1 \leq i_1 < i_2 < \dots < i_{|A|} \leq n$ and $1 \leq j_1 < j_2 < \dots < j_{|A|} \leq m$. An alignment A is called *structural* iff $|A| = m$ and $i_m = i_1 + m - 1$. That is, all sequences in \mathbf{Y} are matched, and the matched positions in \mathbf{X} are contiguous. The set of all alignments is denoted as $\mathcal{A}_{n,m}$, and the set of all structural alignments is denoted as $\mathcal{S}_{n,m}$.

The *cost* of an alignment A (with respect to \mathbf{X} and \mathbf{Y}) is defined as:

$$\delta(A) = \sum_{(i,j) \in A} \delta_{\text{LCS}}(X_i, Y_j) + (m - |A|) \max_{i,j} \delta_{\text{LCS}}(X_i, Y_j).$$

That is, unaligned parts of \mathbf{Y} are penalized by $\max_{i,j} \delta_{\text{LCS}}(X_i, Y_j)$.

Given a sequence X , the *type* of X is defined as $(|X|, \sum_i X[i])$, where each letter is assumed to be a number. Note that when the alphabet is simply $\{0, 1\}$, $\sum_i X[i]$ is simply the number of occurrences of 1 in X .

The following key lemma was proved in [8] (Lemma 4.3 of [8]):

Lemma 1 ([8]). Let $\mathbf{X} = (X_1, \dots, X_n)$ and $\mathbf{Y} = (Y_1, \dots, Y_m)$ be two linearly ordered sets of binary strings such that $n \geq m$, all X_i are of type $\mathcal{T}_X = (\ell_X, s_X)$, and all Y_i are of type $\mathcal{T}_Y = (\ell_Y, s_Y)$. There are two binary strings $S_X = \text{GA}_X^{m, \mathcal{T}_Y}(X_1, \dots, X_n)$, $S_Y = \text{GA}_Y^{n, \mathcal{T}_X}(Y_1, \dots, Y_m)$ and an integer C meeting the following requirements:

- $\min_{A \in \mathcal{A}_{n,m}} \delta(A) \leq \delta_{\text{LCS}}(S_X, S_Y) - C \leq \min_{A \in \mathcal{S}_{n,m}} \delta(A)$.
- The types of S_X, S_Y and the integer C only depend on $n, m, \mathcal{T}_X, \mathcal{T}_Y$.
- S_X, S_Y , and C can be calculated in time $\mathcal{O}((n+m)(\ell_X + \ell_Y))$ (hence $|S_X|$ and $|S_Y|$ are both $\mathcal{O}((n+m)(\ell_X + \ell_Y))$).

Note that the term GA comes from the word gadget.

Intuitively, computing an optimal alignment (or an optimal structural alignment) of two sets of sequences is at least as hard as computing a longest common subsequence. The above lemma gives a reduction from the computation of a number s with $\min_{A \in \mathcal{A}_{n,m}} \delta(A) \leq s \leq \min_{A \in \mathcal{S}_{n,m}} \delta(A)$ (which can be regarded as an approximation of optimal alignments) to a single LCS instance.

We will use the above lemma as a black box to devise two encodings, the clique node gadget $\text{CNG}(t)$ and the clique list gadget $\text{CLG}(t)$, for a k -clique t in a graph in such a way that we can decide whether two k -cliques t_1, t_2 form a $2k$ -clique according the value of $\delta_{\text{LCS}}(\text{CNG}(t_1), \text{CLG}(t_2))$.

When invoking the lemma, \mathbf{X}, \mathbf{Y} are designed in such a way that we can test whether a condition is met (e.g. whether two given k -cliques form a $2k$ -clique) by the value of $\min_{A \in \mathcal{A}_{n,m}} \delta(A)$. We will show that $\min_{A \in \mathcal{A}_{n,m}} \delta(A) = \min_{A \in \mathcal{S}_{n,m}} \delta(A)$ for the case we are interested in. Therefore, we can infer whether the condition we are interested in is met from the value of $\delta_{\text{LCS}}(S_X, S_Y)$.

3. From Cliques to RNA Folding

The goal of this section is to prove Theorem 2.

Let $G = (V, E)$ be a graph, and let $n = |V|$. We write \mathcal{C}_k to denote the set of k -cliques in G . We fix $\Sigma = \{0, 1\}$. As in [3], we will construct a sequence $S_G \in (\Sigma \cup \Sigma')^*$ such that we can decide whether G has a $3k$ -clique according to the value of $\text{RNA}(S_G)$.

As our framework of the construction of S_G is similar to the one in [3], we will give the building blocks (for constructing S_G) the same names as their analogues in [3], despite that they may have different lower-level implementations.

The high-level plan is described as following:

In Section 3.1 we describe two encodings $\text{CNG}(t), \text{CLG}(t)$ for a k -clique t based on the black box described in Lemma 1. In Section 3.2, adapting the encodings shown in the previous subsection as the building blocks, we present the definition of the binary sequence S_G . We will give a lower bound on $\text{RNA}(S_G)$ by demonstrating an RNA folding of S_G , and the bound will depend on whether G has a $3k$ -clique.

The goal of the next two subsections is to show that the bound given in Section 3.2 is actually the exact value of $\text{RNA}(S_G)$. In Section 3.3, we show that there exists an optimal RNA folding of S_G meeting several constraints. These constraints will simplify the calculation of $\text{RNA}(S_G)$, and we will work out the exact calculation in Section 3.4.

3.1. Testing $2k$ -cliques via LCS

We associate each vertex $v \in V$ a distinct integer in $\{0, 1, \dots, n-1\}$. Let s_v be the binary encoding of such integer with $|s_v| = \lceil \log(n) \rceil$. We define \bar{v} to be the binary string resulted by replacing each

0 in s_v with 01 and replacing each 1 in s_v with 10. It is clear that for each $v \in V$, \bar{v} is of type $\mathcal{T}_0 = (2\lceil \log(n) \rceil, \lceil \log(n) \rceil)$, and $\delta_{\text{LCS}}(\bar{u}, \bar{v}) = 0$ iff $u = v$.

In this subsection we present two encodings $\text{CNG}(t)$, $\text{CLG}(t)$ for a k -clique t such that we can infer whether two k -cliques t_1, t_2 form a $2k$ -clique from the value of $\delta_{\text{LCS}}(\text{CNG}(t_1), \text{CLG}(t_2))$.

For each $v \in V$, the *list gadget* $\text{LG}(v)$ and the *node gadget* $\text{NG}(v)$ are defined as following:

- $\text{LG}(v) = \text{GA}_X^{1, \mathcal{T}_0}(\bar{u}_1, \bar{u}_2, \dots, \bar{u}_{|N(v)|}, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}, \dots, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil})$, where $N(v) = \{u_1, u_2, \dots, u_{|N(v)|}\}$, and the number of occurrences of $1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}$ is $n - |N(v)|$.
- $\text{NG}(v) = \text{GA}_Y^{n, \mathcal{T}_0}(\bar{v})$.

Lemma 2. *There is a constant c_0 , depending only on n , such that for any $v_1, v_2 \in V$, we have $\{v_1, v_2\} \in E$ iff $\delta_{\text{LCS}}(\text{LG}(v_1), \text{NG}(v_2)) = c_0 = \min_{v'_1, v'_2 \in V} \delta_{\text{LCS}}(\text{LG}(v'_1), \text{NG}(v'_2))$.*

Proof. We let $N(v_1) = \{u_1, u_2, \dots, u_{|N(v_1)|}\}$.

Let $\mathbf{X} = (\bar{u}_1, \bar{u}_2, \dots, \bar{u}_{|N(v_1)|}, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}, \dots, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil})$, where the number of occurrences of $1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}$ is $n - |N(v_1)|$, and let $\mathbf{Y} = (\bar{v}_2)$.

In view of Lemma 1, we have $\min_{A \in \mathcal{A}_{n,1}} \delta(A) \leq \delta_{\text{LCS}}(\text{LG}(v_1), \text{NG}(v_2)) - C \leq \min_{A \in \mathcal{S}_{n,1}} \delta(A)$, for some C whose value depends on $|\mathbf{X}|, |\mathbf{Y}|$, and \mathcal{T}_0 . As these parameters depend solely on n , the number C only depends on n .

Since $|\mathbf{Y}| = 1$, any non-empty alignment between \mathbf{X} and \mathbf{Y} is structural. This implies that $\delta_{\text{LCS}}(\text{LG}(v_1), \text{NG}(v_2)) - C = \min_{A \in \mathcal{A}_{n,1}} \delta(A) = \min_{A \in \mathcal{S}_{n,1}} \delta(A)$.

When $\{v_1, v_2\} \in E$, since \bar{v}_2 is contained in \mathbf{X} , clearly $\min_{A \in \mathcal{S}_{n,m}} \delta(A) = 0$. When $\{v_1, v_2\} \notin E$, \bar{v}_2 does not appear in \mathbf{X} , so $\min_{A \in \mathcal{S}_{n,m}} \delta(A) > 0$. Note that $1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil} \neq \bar{v}$, for any $v \in V$.

As a result, $\{v_1, v_2\} \in E$ iff $\delta_{\text{LCS}}(\text{LG}(v_1), \text{NG}(v_2)) = C = \min_{v'_1, v'_2 \in V} \delta_{\text{LCS}}(\text{LG}(v'_1), \text{NG}(v'_2))$. Hence setting $c_0 = C$ suffices. \square

We let \mathcal{T}_X be the type of the list gadgets, and we let \mathcal{T}_Y be the type of the node gadgets. For each k -clique $t = \{u_1, u_2, \dots, u_k\}$, we define the *clique node gadget* $\text{CNG}(t)$ and the *clique list gadget* $\text{CLG}(t)$ as following:

- $\text{CLG}(t) = \text{GA}_X^{k^2, \mathcal{T}_Y}(\text{LG}(u_1), \dots, \text{LG}(u_1), \text{LG}(u_2), \dots, \text{LG}(u_2), \dots, \text{LG}(u_k), \dots, \text{LG}(u_k))$, where the number of occurrences of each $\text{LG}(u_i)$ is k .
- $\text{CNG}(t) = \text{GA}_Y^{k^2, \mathcal{T}_X}(\text{NG}(u_1), \text{NG}(u_2), \dots, \text{NG}(u_k), \text{NG}(u_1), \text{NG}(u_2), \dots, \text{NG}(u_k), \dots, \text{NG}(u_1), \text{NG}(u_2), \dots, \text{NG}(u_k))$, where the number of occurrences of each $\text{NG}(u_i)$ is k .

We are ready to prove the main lemma in the subsection:

Lemma 3. *There is a constant c_1 , depending only on n, k , such that for any $t_1, t_2 \in \mathcal{C}_k$, $t_1 \cup t_2$ is a $2k$ -clique iff $\delta_{\text{LCS}}(\text{CLG}(t_1), \text{CNG}(t_2)) = c_1 = \min_{t'_1, t'_2 \in \mathcal{C}_k} \delta_{\text{LCS}}(\text{CLG}(t'_1), \text{CNG}(t'_2))$.*

Proof. Let $t_1 = \{u_1, u_2, \dots, u_k\}$, and let $t_2 = \{v_1, v_2, \dots, v_k\}$.

Let $\mathbf{X} = (\text{LG}(u_1), \dots, \text{LG}(u_1), \text{LG}(u_2), \dots, \text{LG}(u_2), \dots, \text{LG}(u_k), \dots, \text{LG}(u_k))$, where each $\text{LG}(u_i)$ appears k times, and let $\mathbf{Y} = (\text{NG}(v_1), \text{NG}(v_2), \dots, \text{NG}(v_k), \text{NG}(v_1), \text{NG}(v_2), \dots, \text{NG}(v_k), \dots, \text{NG}(v_1), \text{NG}(v_2), \dots, \text{NG}(v_k))$, where each $\text{NG}(v_i)$ appears k times.

In view of Lemma 2, we have $\min_{w_1, w_2 \in V} \delta_{\text{LCS}}(\text{LG}(w_1), \text{NG}(w_2)) \geq c_0$, so we can lower bound $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A)$ by $k^2 c_0$.

If $\max_{i,j} \delta_{\text{LCS}}(X_i, Y_j) = c_0$, any alignment has cost $k^2 c_0$. When $\max_{i,j} \delta_{\text{LCS}}(X_i, Y_j) > c_0$, it is easy to observe that in order to achieve $\delta(A) = k^2 c_0$, all sequences in \mathbf{Y} must be aligned (as the

cost for any unaligned sequence in \mathbf{Y} is now $> c_0$). Therefore, any alignment A with $\delta(A) = k^2 c_0$ must be $A = \{(i, i) | i \in \{1, 2, \dots, k^2\}\}$ with $\delta_{\text{LCS}}(X_i, Y_i) = c_0$, for all $i \in \{1, 2, \dots, k^2\}$.

In view of the above, $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A) = k^2 c_0$ iff $\delta_{\text{LCS}}(X_i, Y_i) = c_0$ for all $i \in \{1, 2, \dots, k^2\}$.

Since $A = \{(i, i) | i \in \{1, 2, \dots, k^2\}\}$ is structural, $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A) = k^2 c_0$ iff $\min_{A \in \mathcal{S}_{k^2, k^2}} \delta(A) = k^2 c_0$. Therefore, in view of Lemma 1, there exists a constant C such that:

- If $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A) = k^2 c_0$, then $\delta_{\text{LCS}}(\text{CLG}(t_1), \text{CNG}(t_2)) = k^2 c_0 + C$.
- If $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A) > k^2 c_0$, then $\delta_{\text{LCS}}(\text{CLG}(t_1), \text{CNG}(t_2)) > k^2 c_0 + C$.

Moreover, the value of C depends only on $|\mathbf{X}|, |\mathbf{Y}|, \mathcal{T}_X, \mathcal{T}_Y$. As these parameters depend solely on n, k , the number C only depends on n, k .

When $t_1 \cup t_2$ is a $2k$ -clique, all vertices in t_1 are adjacent to all vertices in t_2 . In view of Lemma 2, $\forall_{i,j} \delta_{\text{LCS}}(X_i, Y_j) = c_0$. Hence $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A) = k^2 c_0$, implying that $\delta_{\text{LCS}}(\text{CLG}(t_1), \text{CNG}(t_2)) = k^2 c_0 + C$.

When $t_1 \cup t_2$ is not a $2k$ -clique, there exist $u_i \in t_1, v_j \in t_2$ such that $\{u_i, v_j\} \notin E$. According to our definition of \mathbf{X} and \mathbf{Y} , we have $X_{j+k(i-1)} = \text{LG}(u_i)$, $Y_{j+k(i-1)} = \text{NG}(v_j)$, and hence $\delta_{\text{LCS}}(X_{j+k(i-1)}, Y_{j+k(i-1)}) > c_0$. This implies that $\min_{A \in \mathcal{A}_{k^2, k^2}} \delta(A) > k^2 c_0$, which leads to $\delta_{\text{LCS}}(\text{CLG}(t_1), \text{CNG}(t_2)) > k^2 c_0 + C$.

As a result, $t_1 \cup t_2$ is a $2k$ -clique iff $\delta_{\text{LCS}}(\text{CLG}(t_1), \text{CNG}(t_2)) = k^2 c_0 + C = \min_{t'_1, t'_2 \in \mathcal{C}_k} \delta_{\text{LCS}}(\text{CLG}(t'_1), \text{CNG}(t'_2))$. Setting $c_1 = k^2 c_0 + C$ suffices. \square

The following lemma is a simple consequence of Lemma 1:

Lemma 4. *There exist four integers $\ell_{\text{CNG},0}, \ell_{\text{CNG},1}, \ell_{\text{CLG},0}, \ell_{\text{CLG},1} \in \mathcal{O}(k^2 n \log(n))$, such that for any $t \in \mathcal{C}_k$,*

- $\ell_{\text{CNG},b}$ = the number of occurrences of b in $\text{CNG}(t)$, $b \in \{0, 1\}$.
- $\ell_{\text{CLG},b}$ = the number of occurrences of b in $\text{CLG}(t)$, $b \in \{0, 1\}$.

Proof. As a consequence of Lemma 1, all $\text{CNG}(t)$ have the same type, and all $\text{CLG}(t)$ have the same type. Therefore, the existence of these four integers is guaranteed.

In view of Lemma 1, for all $v \in V$, both $\text{LG}(v)$ and $\text{NG}(v)$ have length at most $(n+1) \cdot (2\lceil \log(n) \rceil + 2\lfloor \log(n) \rfloor) = \mathcal{O}(n \log(n))$. Applying Lemma 1 again, the length of both $\text{CNG}(t)$ and $\text{CLG}(t)$ for all $t \in \mathcal{C}_k$ is $(k^2 + k^2)(\mathcal{O}(n \log(n)) + \mathcal{O}(n \log(n))) = \mathcal{O}(k^2 n \log(n))$.

As a result, the four integers can be bounded by $\mathcal{O}(k^2 n \log(n))$. \square

3.2. The RNA sequence S_G

Based on the parameters in Lemma 4, we define $\ell_0 = \ell_{\text{CNG},0} + \ell_{\text{CNG},1} + \ell_{\text{CLG},0} + \ell_{\text{CLG},1} = \mathcal{O}(k^2 n \log(n))$; for $i \in \{1, 2, 3\}$, we set $\ell_i = 100\ell_{i-1}$; and $\ell_4 = 100|\mathcal{C}_k|\ell_3 = \mathcal{O}(k^2 n^{k+1} \log(n))$.

The RNA sequence S_G is then defined as following:

$$S_G = 0^{\ell_4} \left[0^{\ell_3} \bigcirc_{t \in \mathcal{C}_k} \left(\text{CG}_\alpha(t) 0^{\ell_3} \right) \right] 0^{\ell_4} \left[0^{\ell_3} \bigcirc_{t \in \mathcal{C}_k} \left(\text{CG}_\beta(t) 0^{\ell_3} \right) \right] 0^{\ell_4} \left[0^{\ell_3} \bigcirc_{t \in \mathcal{C}_k} \left(\text{CG}_\gamma(t) 0^{\ell_3} \right) \right],$$

where

$$\begin{aligned} \text{CG}_\alpha(t) &= 1'^{2\ell_2} p(\text{CLG}(t)^R) 0'^{\ell_1} 1^{\ell_2} 0^{\ell_1} \text{CNG}(t) 1^{\ell_2}, \\ \text{CG}_\beta(t) &= 1'^{\ell_2} p(\text{CLG}(t)^R) 0'^{\ell_1} 1'^{2\ell_2} 0'^{\ell_1} p(\text{CNG}(t)) 1'^{\ell_2}, \\ \text{CG}_\gamma(t) &= 1^{\ell_2} \text{CLG}(t)^R 0^{\ell_1} 1^{\ell_2} 0^{\ell_1} \text{CNG}(t) 1^{2\ell_2}. \end{aligned}$$

For any $t \in \mathcal{C}_k$, $x \in \{\alpha, \beta, \gamma\}$, the string $\text{CG}_x(t)$ is called a *clique gadget*.

Note that $\text{CG}_\alpha(t) \in (\Sigma \cup \Sigma')^*$, $\text{CG}_\beta(t) \in \Sigma'^*$, and $\text{CG}_\gamma(t) \in \Sigma^*$.

It is obvious that $|S_G| = \mathcal{O}(|\mathcal{C}_k| \ell_0) = \mathcal{O}(k^2 n^{k+1} \log(n))$.

Before proceeding further, we explain some intuitions behind the definition of S_G and give a simple lower bound on $\text{RNA}(S_G)$ by constructing an RNA folding as following:

- The pairings between letters in some $0'^{\ell_3}$ and some 0^{ℓ_4} sometimes make a clique gadget unable to participate in the RNA folding with other clique gadgets. In this sense, a clique gadget is said to be “blocked” if the letters within the clique gadget only pair up with other letters within the same clique gadget or some 0 in a 0^{ℓ_4} .
Let’s try linking all the $0'$ in all $0'^{\ell_3}$ to some 0 in some 0^{ℓ_4} in such a way that all clique gadgets are blocked except $\text{CG}_\alpha(t_\alpha)$, $\text{CG}_\beta(t_\beta)$, and $\text{CG}_\gamma(t_\gamma)$. This gives us $3(|\mathcal{C}_k| + 1)\ell_3$ amount of pairs. See Fig. 1.
- For a clique gadget that is “blocked”, our design of S_G ensures that the number of pairs involving letters in the clique gadget (in certain optimal RNA foldings) is irrelevant to its corresponding k -clique (we will prove it later):

- For a blocked $\text{CG}_\alpha(t)$, since ℓ_2 is significantly larger than ℓ_1, ℓ_0 , an optimal way to pair up the letters is to match as many $\{1', 1\}$ as possible. This gives us $\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1})$ pairs.
- For a blocked $\text{CG}_\beta(t)$, since we do not have any 1 here, the best we can do is to match all $0'$ to some 0^{ℓ_4} . This gives us $2\ell_1 + \ell_{\text{CLG},0} + \ell_{\text{CNG},0}$ pairs.
- For a blocked $\text{CG}_\gamma(t)$, no matching can be made.

Therefore, the total amount of pairs involving blocked clique gadgets is $(|\mathcal{C}_k| - 1)(2\ell_1 + 2\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1}) + \ell_{\text{CLG},0} + \ell_{\text{CNG},0})$. See Fig. 2 for an illustration.

- For the three clique gadgets that are not blocked, we will later see that (in certain optimal RNA foldings) $\text{CG}_\alpha(t_\alpha), \text{CG}_\beta(t_\beta), \text{CG}_\gamma(t_\gamma)$ correspond to a $3k$ -clique if the graph has one. It is a simple exercise to construct an RNA folding for $\text{CG}_\alpha(t_\alpha) \circ \text{CG}_\beta(t_\beta) \circ \text{CG}_\gamma(t_\gamma)$ that uses up all the $1'^{2\ell_2}, 1^{2\ell_2}, 1'^{\ell_2}, 1^{\ell_2}, 0'^{\ell_1}, 0^{\ell_1}$ and has cardinality $6\ell_2 + 3\ell_1 + \frac{1}{2}(\ell_0 - \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\beta))) + \frac{1}{2}(\ell_0 - \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\gamma))) + \frac{1}{2}(\ell_0 - \delta_{\text{LCS}}(\text{CLG}(t_\beta), \text{CNG}(t_\gamma)))$. Recall that $\frac{1}{2}(\ell_0 - \delta_{\text{LCS}}(\text{CLG}(t_x), \text{CNG}(t_y)))$ is the length of the LCS between $\text{CLG}(t_x)$ and $\text{CNG}(t_y)$. See Fig. 3 for an illustration.

In light of the above discussion, we define:

- $m_1 = 3(|\mathcal{C}_k| + 1)\ell_3 + (|\mathcal{C}_k| - 1)(2\ell_1 + 2\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1}) + \ell_{\text{CLG},0} + \ell_{\text{CNG},0})$,
- $m_2 = 6\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - \min_{t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k} \frac{1}{2}(\delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\beta)) + \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\gamma)) + \delta_{\text{LCS}}(\text{CLG}(t_\beta), \text{CNG}(t_\gamma)))$.

The next lemma, which gives a lower bound on $\text{RNA}(S_G)$, is then implied instantly by the above discussion.

Lemma 5. $\text{RNA}(S_G) \geq m_1 + m_2$.

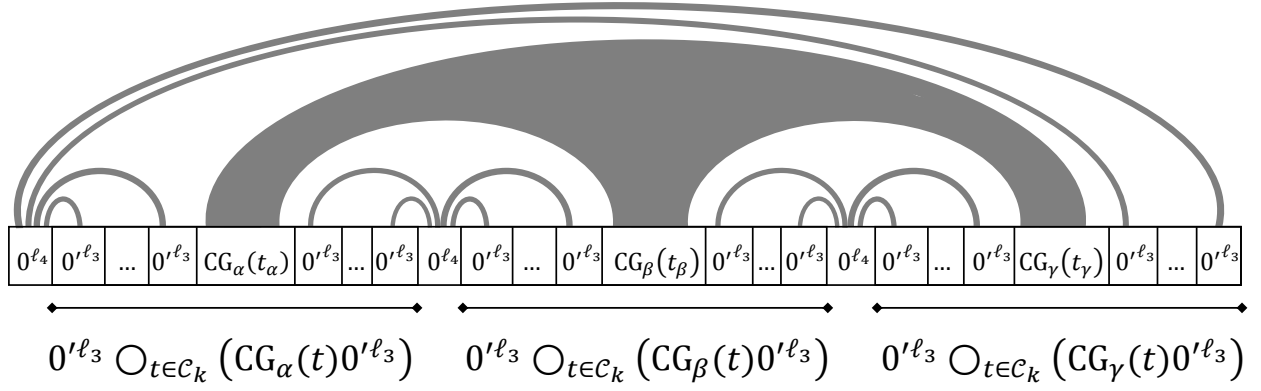


Fig. 1. The three selected clique gadgets and the matchings between $0'^{\ell_3}$ and 0^{ℓ_4} .

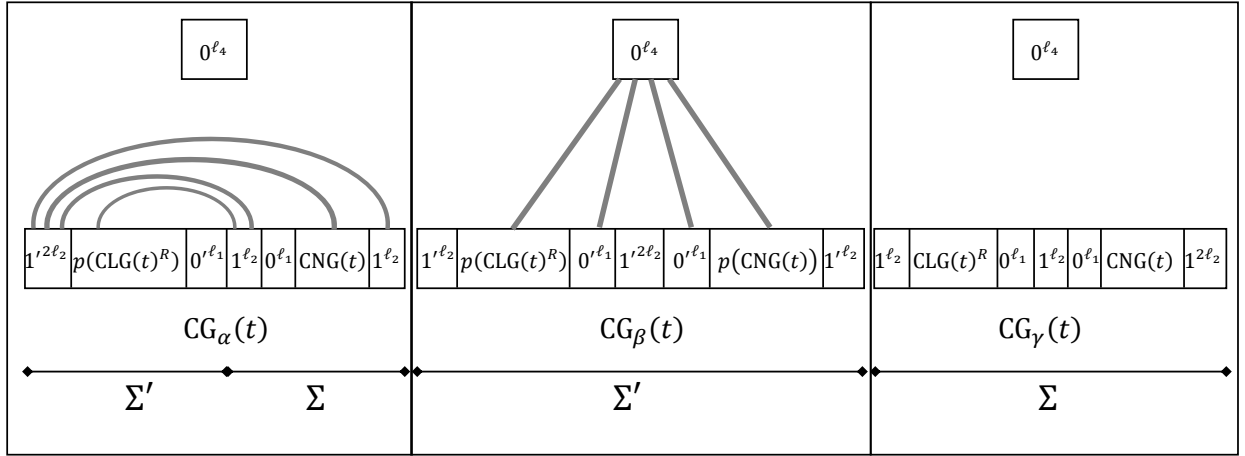


Fig. 2. The matchings between a blocked clique gadget and 0^{ℓ_4} .

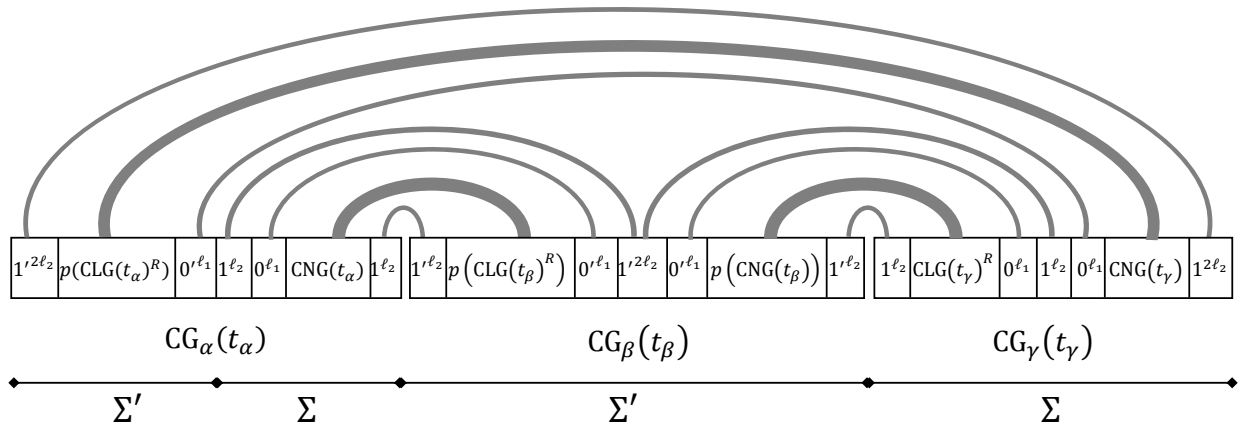


Fig. 3. The matchings within the three selected clique gadgets.

Ultimately we will show that $\text{RNA}(S_G) = m_1 + m_2$, and clearly this offers enough information for us to decide whether G has a $3k$ -clique.

The following lemma calculates $\text{RNA}(\cdot)$ of some sequences, and they will be useful in the subsequent discussion.

Lemma 6. *The following statements are true for any $t, t' \in \mathcal{C}_k$:*

1. $\text{RNA}(0^{\ell_4} CG_\alpha(t)) = 2\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1})$
2. $\text{RNA}(0^{\ell_4} CG_\beta(t)) = 2\ell_1 + \ell_{\text{CLG},0} + \ell_{\text{CNG},0}$
3. $\text{RNA}(0^{\ell_4} CG_\gamma(t)) = 0$
4. $\text{RNA}(0^{\ell_4} CG_\alpha(t)0^{\ell_4} CG_\beta(t')) \leq 3.1\ell_1 + 2\ell_2$
5. $\text{RNA}(0^{\ell_4} CG_\alpha(t)0^{\ell_4} CG_\gamma(t')) \leq 1.1\ell_1 + 2\ell_2$
6. $\text{RNA}(0^{\ell_4} CG_\beta(t)0^{\ell_4} CG_\gamma(t')) \leq 1.1\ell_1 + 4\ell_2$

Proof. The value of $\text{RNA}(\cdot)$ for each of the six sequences are calculated as following:

1. Linking as many 1 to $1'$ gets a matching of size $m = 2\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1})$. To see that it is optimal, it suffices to show that both $(0', 0)$ and $(0, 0')$ cannot appear in an optimal RNA folding:
 - If the RNA folding contains $(0, 0')$, then none of $1'$ can participate in the RNA folding. As the total number of $0'$ is $\ell_1 + \ell_{\text{CLG},0}$, the size of RNA folding is at most $\ell_1 + \ell_{\text{CLG},0} < m$.
 - If the RNA folding contains $(0', 0)$, then at most $\ell_{\text{CLG},1}$ number of letters within the middle 1^{ℓ_2} (the one between $0'^{\ell_1}$ and 0^{ℓ_1}) can participate in the RNA folding. It implies that the number of $(1', 1)$ pairs in the RNA folding is at most $\ell_{\text{CLG},1} + \ell_2$. Hence the size of the RNA folding can be upper bounded by $(\ell_1 + \ell_{\text{CLG},0}) + (\ell_{\text{CLG},1} + \ell_2) < m$.
2. Since there is no 1, the equation follows from the fact that there are $2\ell_1 + \ell_{\text{CLG},0} + \ell_{\text{CNG},0}$ occurrences of $0'$, all of which can be matched to some 0 without crossing.
3. No matching can be made since there is no $0', 1'$.
4. The value of $\text{RNA}(\cdot)$ can be upper bounded by the number of 1 and $0'$. This is $(2\ell_2 + \ell_{\text{CNG},1}) + (3\ell_1 + 2\ell_{\text{CLG},0} + \ell_{\text{CNG},0}) \leq 3.1\ell_1 + 2\ell_2$.
5. The value of $\text{RNA}(\cdot)$ can be upper bounded by the number of $1'$ and $0'$. This is $(2\ell_2 + \ell_{\text{CLG},1}) + (\ell_1 + \ell_{\text{CLG},0}) \leq 1.1\ell_1 + 2\ell_2$.
6. We define $S = 0^{\ell_4} \circ (1^{\ell_2} 0'^{\ell_1} 1'^{2\ell_2} 0'^{\ell_1} 1'^{\ell_2}) \circ 0^{\ell_4} \circ (1^{\ell_2} 0^{\ell_1} 1^{\ell_2} 0^{\ell_1} 1^{2\ell_2})$, which is the result of removing the clique node gadgets and the clique list gadgets in $0^{\ell_4} CG_\beta(t)0^{\ell_4} CG_\gamma(t')$. It is clear that $\text{RNA}(0^{\ell_4} CG_\beta(t)0^{\ell_4} CG_\gamma(t')) \leq 0.1\ell_1 + \text{RNA}(S)$, as the total length of the removed substrings can be upper bounded by $0.1\ell_1$. Therefore, it suffices to show that $\text{RNA}(S) \leq \ell_1 + 4\ell_2$.

Let A be any RNA folding of S :

- Case: there are some $(0, 0') \in A$ where the $0'$ comes from the first $0'^{\ell_1}$ in S . Clearly, the first substring $1'^{\ell_2}$ cannot participate in any pairing. Therefore, $|A| \leq |0'^{\ell_1} 1'^{2\ell_2} 0'^{\ell_1} 1'^{\ell_2}| = 2\ell_1 + 3\ell_2 < \ell_1 + 4\ell_2$.
- Case: there are some $(0', 0) \in A$ where the $0'$ comes from the first $0'^{\ell_1}$ in S . In this situation, at most half of the $1^{2\ell_2}$ can participate in the RNA folding, since only the first $1'^{\ell_2}$ in S is reachable from $1^{2\ell_2}$ without crossing a pair $(0', 0)$. Therefore, $|A|$ is at most the total number of $0'$ and 1 in S minus ℓ_2 , i.e. $|A| \leq 2\ell_1 + 3\ell_2 < \ell_1 + 4\ell_2$.
- Case: the first $0'^{\ell_1}$ in S does not participate in the RNA folding. Then, $|A| \leq |1'^{\ell_2} 1'^{2\ell_2} 0'^{\ell_1} 1'^{\ell_2}| = \ell_1 + 4\ell_2$.

□

Note that (1), (2), (3) in Lemma 6 imply that the RNA folding for blocked clique gadgets described in Fig. 2 is optimal, and the optimal number of pairings is irrelevant to the corresponding k -clique.

3.3. Optimal RNA foldings of S_G

In the previous subsection, we describe an RNA folding of S_G containing $m_1 + m_2$ pairs. The two key properties of this RNA folding are:

Property 1. All $0'$ in all $0'^{\ell_3}$ are paired up with some 0 in some 0^{ℓ_4} .

Property 2. All clique gadgets are “blocked” by the pairings between $0'^{\ell_3}$ and 0^{ℓ_4} , except the three clique gadgets: $\text{CG}_\alpha(t_\alpha), \text{CG}_\beta(t_\beta), \text{CG}_\gamma(t_\gamma)$, for some $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$.

The goal in this section is to show that there is an optimal RNA folding having the above two properties, which facilitates the calculation of $\text{RNA}(S_G)$ in the next subsection.

Lemma 7. *For any RNA folding A of S_G , if there exists a pair linking a $0'$ in a specific $0'^{\ell_3}$ (denoted as S_1) to a 0 in a specific 0^{ℓ_4} (denoted as S_2), then there exists another RNA folding A' with $|A'| \geq |A|$ where all letters in S_1 are linked to some letters in S_2 .*

Proof. It immediately follows from the fact that ℓ_4 is greater than the total number of $0'$ in S_G . It makes rematching all the letters in S_1 to some letters in S_2 possible. \square

Lemma 8 ensures that there is an optimal RNA folding having Property 1:

Lemma 8. *There is an optimal RNA folding A of S_G having Property 1.*

Proof. Let's choose any RNA folding A of S_G with $|A| = \text{RNA}(S_G)$. In view of Lemma 7, we can assume that for each $0'^{\ell_3}$ in S_G , either all its letters are matched to some 0 in the same 0^{ℓ_4} or none of its letters is matched to any 0 in any 0^{ℓ_4} . Let z denote the number of $0'^{\ell_3}$ such that none of its letters are matched to any 0 in any 0^{ℓ_4} .

For some $t \in \mathcal{C}_k$, and for some $x \in \{\alpha, \beta, \gamma\}$, $\text{CG}_x(t)$ is said to be “trapped” in A if all letters within $\text{CG}_x(t)$ are either unmatched, matched to letters within $\text{CG}_x(t)$, or matched to letters in some 0^{ℓ_4} .

We note that a sufficient condition for $\text{CG}_x(t)$ to be trapped is that the letters in its two neighboring $0'^{\ell_3}$ are all matched to the same 0^{ℓ_4} . The cases that the condition is violated is enumerated as follows:

1. The two neighboring $0'^{\ell_3}$ of $\text{CG}_x(t)$ are matched to different 0^{ℓ_4} , and this occurs at most $2|\{\alpha, \beta, \gamma\}| = 6$ times (i.e. at most two times per $x \in \{\alpha, \beta, \gamma\}$).
2. A neighboring $0'^{\ell_3}$ of $\text{CG}_x(t)$ is not matched to any 0^{ℓ_4} , and this occurs at most $2z$ times.

Therefore, the number of clique gadgets that are not trapped in A is at most $6 + 2z$.

Using this information, we can derive an upper bound of $|A|$:

$$\begin{aligned}
|A| &\leq (3(|\mathcal{C}_k| + 1) - z)\ell_3 && \text{(matched } 0'^{\ell_3}) \\
&+ |\mathcal{C}_k| \left(\max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\alpha(t)) + \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\beta(t)) \right. && \text{(trapped clique gadgets)} \\
&\quad \left. + \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\gamma(t)) \right) \\
&+ (6 + 2z) \max_{t \in \mathcal{C}_k, x \in \{\alpha, \beta, \gamma\}} |\text{CG}_x(t)|. && \text{(remaining clique gadgets)}
\end{aligned}$$

In view of the calculation in Lemma 6, $|A|$ is at most

$$m_1 - z\ell_3 + (2\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1}) + 2\ell_1 + \ell_{\text{CLG},0} + \ell_{\text{CNG},0}) + (6 + 2z) \max_{t,x} |\text{CG}_x(t)|.$$

Since $2\ell_2 + \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1}) + 2\ell_1 + \ell_{\text{CLG},0} + \ell_{\text{CNG},0} < 0.1\ell_3$, and since the length of a clique gadget $< 0.1\ell_3$, we have:

$$|A| < m_1 - 0.8z\ell_3 + 0.7\ell_3.$$

Therefore, $|A| < m_1 < \text{RNA}(S_G)$ if $z > 0$. Hence we must have $z = 0$, i.e. all $0'$ in all $0'^{\ell_3}$ are paired up with some 0 in some 0^{ℓ_4} . \square

To proceed further, some terminologies are needed to formally define the Property 2:

Definition 1. Let A be an RNA folding of a sequence where S_1, S_2 are two substrings (subsequences of consecutive elements). We write $S_1 \xleftrightarrow{A} S_2$ iff

- there exists $\{x_1, x_2\} \in A$ with $x_1 \in S_1, x_2 \in S_2$.
- S_1, S_2 are disjoint substrings.

For example, “ $\text{CG}_x(t_1)$ is blocked in A ” is equivalent to “there exist no $y \in \{\alpha, \beta, \gamma\}, t_2 \in \mathcal{C}_k$ such that $\text{CG}_x(t_1) \xleftrightarrow{A} \text{CG}_y(t_2)$ ”.

Definition 2. \mathcal{M}_α is defined as the set of RNA foldings of S_G such that $A \in \mathcal{M}_\alpha$ iff

- A has Property 1, and
- there exist $t_{\alpha,1}, t_{\alpha,2}, t_\beta, t_\gamma \in \mathcal{C}_k$ such that $t_{\alpha,1} \neq t_{\alpha,2}$, and for any $t_1, t_2 \in \mathcal{C}_k, \{u_1, u_2\} \subseteq \{\alpha, \beta, \gamma\}$, $\text{CG}_{u_1}(t_1) \xleftrightarrow{A} \text{CG}_{u_2}(t_2)$ implies that $\{(u_1, t_1), (u_2, t_2)\} \in \{(\alpha, t_{\alpha,1}), (\beta, t_\beta), (\alpha, t_{\alpha,2}), (\gamma, t_\gamma)\}$.

\mathcal{M}_β and \mathcal{M}_γ are defined analogously.

Definition 3. $\mathcal{M}_{\alpha,\beta,\gamma}$ is defined as the set of RNA foldings of S_G such that $A \in \mathcal{M}_{\alpha,\beta,\gamma}$ iff

- A has Property 1, and
- there exist $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$ such that for any $t_1, t_2 \in \mathcal{C}_k, \{u_1, u_2\} \subseteq \{\alpha, \beta, \gamma\}$, $\text{CG}_{u_1}(t_1) \xleftrightarrow{A} \text{CG}_{u_2}(t_2)$ implies that $\{(u_1, t_1), (u_2, t_2)\} \subseteq \{(\alpha, t_\alpha), (\beta, t_\beta), (\gamma, t_\gamma)\}$.

Using the above notions, it is clear that $A \in \mathcal{M}_{\alpha,\beta,\gamma}$ iff A has both Property 1 and Property 2. In the remainder of the subsection, we will prove that there exists an optimal RNA folding of S_G that belongs to $\mathcal{M}_{\alpha,\beta,\gamma}$.

To ease the notation, for each $x \in \{\alpha, \beta, \gamma\}$, we call $\text{CG}_x(t)$ an “ x clique gadget”, for all $t \in \mathcal{C}_k$; we write “ C_1 and C_2 are linked (in A)” to denote $C_1 \xleftrightarrow{A} C_2$.

Lemma 9. Let A be an optimal RNA folding of S_G having Property 1. For any $x \in \{\alpha, \beta, \gamma\}$, there does not exist two x clique gadgets C_1, C_2 such that $C_1 \xleftrightarrow{A} C_2$.

Proof. There is a substring $0'^{\ell_3}$ located between C_1 and C_2 . Existence of a pair in A linking a letter in C_1 and a letter in C_2 makes it impossible for any letter in the $0'^{\ell_3}$ be matched to any letter in any 0^{ℓ_4} , a contradiction. \square

Lemma 10. *Let A be an optimal RNA folding of S_G having Property 1. For any $\{x, y\} \in \{\{\alpha, \beta\}, \{\alpha, \gamma\}, \{\beta, \gamma\}\}$, there does not exist two distinct x clique gadgets C_1, C_2 and two (not necessarily distinct) y clique gadgets C_3, C_4 such that $C_1 \xleftrightarrow{A} C_3$ and $C_2 \xleftrightarrow{A} C_4$.*

Proof. Clearly there must be a substring $0'^{\ell_3}$ located between C_1 and C_2 . However, since $C_1 \xleftrightarrow{A} C_3$ and $C_2 \xleftrightarrow{A} C_4$, letters in the substring $0'^{\ell_3}$ can only be matched to letters in C_1, C_2, C_3, C_4 , letters between C_1, C_2 , and letters between C_3, C_4 . This contradicts Property 1. \square

Lemma 11. *Let A be an optimal RNA folding of S_G having Property 1. For any $x \in \{\alpha, \beta, \gamma\}$, suppose that there exist two distinct x clique gadgets C_1, C_2 such that $C_1 \xleftrightarrow{A} C_3$ and $C_2 \xleftrightarrow{A} C_4$ for some clique gadgets C_3, C_4 . Then there does not exist any other pairs of clique gadgets that are linked in A .*

Proof. Let $y, z \in \{\alpha, \beta, \gamma\}$ such that C_3 is a y clique gadget, and C_4 is a z clique gadget. By Lemma 9 and Lemma 10, x, y, z must be distinct.

Suppose that there exist two clique gadgets C_5, C_6 that are linked in A such that $\{C_5, C_6\} \notin \{\{C_1, C_3\}, \{C_2, C_4\}\}$. We show that this leads to a contradiction.

First of all, none of C_5, C_6 can be an x clique gadget. Suppose that C_5 is an x clique gadget. Then by Lemma 9, C_6 is either a y clique gadget or a z clique gadget. In any case, Lemma 10 is violated.

Therefore, we can (without loss of generality) assume that C_5 is a y clique gadget, and C_6 is a z clique gadget.

Since C_1, C_2 are distinct, there must be a substring $0'^{\ell_3}$ located between C_1 and C_2 . Since C_1 is linked to a y gadget, and since C_2 is linked to a z gadget, letters in the substring $0'^{\ell_3}$ can only be paired up with letters in the substring 0^{ℓ_4} bordering both $0'^{\ell_3} \bigcirc_{t \in C_k} (\text{CG}_y(t)0'^{\ell_3})$ and $0'^{\ell_3} \bigcirc_{t \in C_k} (\text{CG}_z(t)0'^{\ell_3})$ (imaging S_G as a circular string). However, the existence of a pair linking a letter in C_5 (an y clique gadget) and a letter in C_6 (a z clique gadget) implies that such 0^{ℓ_4} cannot be reached from the $0'^{\ell_3}$ without a crossing, a contradiction. \square

The following lemma directly follows from Lemma 8 and Lemma 11.

Lemma 12. *There exists an optimal RNA folding of S_G that belongs to $\mathcal{M}_\alpha \cup \mathcal{M}_\beta \cup \mathcal{M}_\gamma \cup \mathcal{M}_{\alpha, \beta, \gamma}$.*

Proof. By Lemma 8, we can restrict our consideration to optimal RNA foldings having Property 1.

Let A be any such optimal RNA folding:

Case 1: For each $x \in \{\alpha, \beta, \gamma\}$, there is at most one x clique gadget that is linked to other clique gadgets. Then $A \in \mathcal{M}_{\alpha, \beta, \gamma}$.

Case 2: For some $x \in \{\alpha, \beta, \gamma\}$, there are two distinct x clique gadgets that are linked to other clique gadgets. By Lemma 11, $A \in \mathcal{M}_x$. \square

We are now in a position to prove the main lemma in the subsection:

Lemma 13. *There exists an optimal RNA folding of S_G that belongs to $\mathcal{M}_{\alpha, \beta, \gamma}$.*

Proof. In view of Lemma 12, it suffices to show that for any $A \in \mathcal{M}_\alpha \cup \mathcal{M}_\beta \cup \mathcal{M}_\gamma$, we have $|A| < \text{RNA}(S_G)$.

Let $t_{x,1}, t_{x,2}, t_y, t_z \in \mathcal{C}_k$ and $\{y, z\} = \{\alpha, \beta, \gamma\} \setminus \{x\}$ be the ones in the definition of \mathcal{M}_x .

We let $A \in \mathcal{M}_x$. Each pair in A falls into one of the following categories:

- The ones linking a $0'$ in some $0'^{\ell_3}$ to a 0 in some 0^{ℓ_4} . There are exactly $3(|\mathcal{C}_k| + 1)\ell_3$ number of such pairs.
- The ones involving a letter in some $\text{CG}_u(t)$, where $(u, t) \notin \{(x, t_{x,1}), (x, t_{x,2}), (y, t_y), (z, t_z)\}$. As any letter in such $\text{CG}_u(t)$ can only be matched to the letters within $\text{CG}_u(t)$ or 0^{ℓ_4} . The number of such pairs can be upper bounded by $(|\mathcal{C}_k| - 2) \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_x(t)) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_y(t)) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_z(t))$.
- The ones involving a letter in some $\text{CG}_u(t)$, where $(u, t) \in \{(x, t_{x,1}), (x, t_{x,2}), (y, t_y), (z, t_z)\}$. The number of such pairs can be upper bounded by $\max_{t, t' \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_x(t) 0^{\ell_4} \text{CG}_y(t')) + \max_{t, t' \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_x(t) 0^{\ell_4} \text{CG}_z(t'))$.

Therefore, using Lemma 6, we can upper bound $|A|$ as following:

- When $x = \alpha$, $|A| \leq m_1 + 2\ell_2 + 4.2\ell_1 - \min(\ell_{\text{CLG},1}, \ell_{\text{CNG},1})$.
- When $x = \beta$, $|A| \leq m_1 + 6\ell_2 + 2.2\ell_1 - \ell_{\text{CLG},0} - \ell_{\text{CNG},0}$.
- When $x = \gamma$, $|A| \leq m_1 + 6\ell_2 + 2.2\ell_1$.

By Lemma 5, we always have $|A| < m_1 + m_2 \leq \text{RNA}(S_G)$ (recall that $m_2 \geq 6\ell_2 + 3\ell_1$). \square

3.4. Calculating $\text{RNA}(S_G)$

In this subsection, we will prove that $\text{RNA}(S_G) = m_1 + m_2$ and finish the proof of Theorem 2.

In view of Lemma 13, when calculating $\text{RNA}(S_G)$, we can restrict our attention to RNA foldings of S_G in $\mathcal{M}_{\alpha,\beta,\gamma}$. Based on the structural property of RNA foldings in $\mathcal{M}_{\alpha,\beta,\gamma}$, we first reduce the calculation of $\text{RNA}(S_G)$ to the calculation of optimal RNA foldings of much simpler sequences.

Lemma 14. $\text{RNA}(S_G) \leq m_1 + \max_{t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma))$.

Proof. In view of Lemma 13, there is an optimal RNA folding of S_G in $\mathcal{M}_{\alpha,\beta,\gamma}$.

For any $A \in \mathcal{M}_{\alpha,\beta,\gamma}$, let $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$ be the ones in the definition of $\mathcal{M}_{\alpha,\beta,\gamma}$. Then, each pair in A falls into one of the following categories:

- The ones linking a $0'$ in some $0'^{\ell_3}$ to a 0 in some 0^{ℓ_4} . There are exactly $3(|\mathcal{C}_k| + 1)\ell_3$ number of such pairs.
- The ones involving a letter in some $\text{CG}_u(t)$, where $(u, t) \notin \{(\alpha, t_\alpha), (\beta, t_\beta), (\gamma, t_\gamma)\}$. As any letter in such $\text{CG}_u(t)$ can only be matched to the letters within $\text{CG}_u(t)$ or 0^{ℓ_4} . The number of such pairs can be upper bounded by $(|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\alpha(t)) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\beta(t)) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \text{RNA}(0^{\ell_4} \text{CG}_\gamma(t))$.
- The ones involving a letter in some $\text{CG}_u(t)$, where $(u, t) \in \{(\alpha, t_\alpha), (\beta, t_\beta), (\gamma, t_\gamma)\}$. The number of such pairs can be upper bounded by $\text{RNA}(0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma))$.

In view of Lemma 6, $|A| = m_1 + \text{RNA}(0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma))$. Hence we conclude the proof. \square

The following auxiliary lemma is useful in the later discussion.

Lemma 15. Let $S = S_1 \circ S_2 \circ S_3 \in \{0, 1, 0', 1'\}^*$, where S_2 is either $11'$ or $1'1$. Then $\text{RNA}(S) = \text{RNA}(S_1 \circ S_3) + 1$.

Proof. It suffices to show that there exists an optimal RNA folding of S such that the 1 and the 1' in S_2 are matched.

We first choose any optimal RNA folding A of S , and then we show that we can modify A in such a way that the 1 and the 1' in S_2 are matched without changing the number of matched pairs.

- Case: the 1 and the 1' in S_2 are already matched. We are done.
- Case: Exactly one of the 1 and the 1' in S_2 is matched. We first unmatched it, and then we pair up the 1 and the 1'. Doing so does not change the number of matched pairs.
- Case: both of the 1 and the 1' in S_2 are matched to some other letters. Let the 1 be matched with x , and let the 1' be matched with y . Removing these two pairs from A and adding $\{x, y\}$ and $\{1, 1'\}$ to A does not change the number of matched pairs.

□

For any choices of three k -cliques $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$, we define:

$$S_{t_\alpha, t_\beta, t_\gamma} = 1^{\ell_2} \circ S_{t_\gamma, t_\alpha} \circ 1^{\ell_2} \circ S_{t_\alpha, t_\beta} \circ 1'^{2\ell_2} \circ S_{t_\beta, t_\gamma},$$

where

$$\begin{aligned} S_{t_\gamma, t_\alpha} &= 0^{\ell_1} \text{CNG}(t_\gamma) p(\text{CLG}(t_\alpha)^R) 0'^{\ell_1}, \\ S_{t_\alpha, t_\beta} &= 0^{\ell_1} \text{CNG}(t_\alpha) p(\text{CLG}(t_\beta)^R) 0'^{\ell_1}, \\ S_{t_\beta, t_\gamma} &= 0'^{\ell_1} p(\text{CNG}(t_\beta)) \text{CLG}(t_\gamma)^R 0^{\ell_1}. \end{aligned}$$

$S_{t_\alpha, t_\beta, t_\gamma}$ is simply a cyclic shift of the concatenation of $\text{CG}_\alpha(t_\alpha)$, $\text{CG}_\beta(t_\beta)$, and $\text{CG}_\gamma(t_\gamma)$ after removing the sequences of 1s and 1's at the beginning and the end of these clique gadgets. The next lemma (together with Lemma 14) reduces the calculation of $\text{RNA}(S_G)$ to the calculation of $\text{RNA}(S_{t_\alpha, t_\beta, t_\gamma})$.

Lemma 16. $\text{RNA}(0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma)) = 4\ell_2 + \text{RNA}(S_{t_\alpha, t_\beta, t_\gamma})$.

Proof. First of all, we state a few easy observations that will be applied in the proof:

- By simply matching only the letters in $\text{CG}_\alpha(t_\alpha)$, $\text{CG}_\beta(t_\beta)$, and $\text{CG}_\gamma(t_\gamma)$ (as described in Fig. 3), we can infer that $\text{RNA}(0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma)) \geq 6\ell_2 + 3\ell_1$.
- The total number of 0' and 1 in $0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma)$ is at most $6\ell_2 + 3.1\ell_1$.
- the difference between the number of 1 and 1' in $0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma)$ is at most $0.1\ell_1$

We claim that in any optimal RNA folding A of $0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma)$, all letters within all 0^{ℓ_4} are not matched:

- Claim: there is no 0' within $\text{CG}_\beta(t_\beta)$ matched to any 0 in the two 0^{ℓ_4} preceding and after $\text{CG}_\beta(t_\beta)$. Recall that $\text{CG}_\beta(t_\beta) = 1'^{\ell_2} p(\text{CLG}(t_\beta)^R) 0'^{\ell_1} 1'^{2\ell_2} 0'^{\ell_1} p(\text{CNG}(t_\beta)) 1'^{\ell_2}$. If there is such a pair, then at least ℓ_2 amount of 1' cannot participate in the RNA folding. Therefore, $|A| \leq (6\ell_2 + 3.1\ell_1) - (\ell_2 - 0.1\ell_1) < \text{RNA}(0^{\ell_4} \text{CG}_\alpha(t_\alpha) 0^{\ell_4} \text{CG}_\beta(t_\beta) 0^{\ell_4} \text{CG}_\gamma(t_\gamma))$.

- Claim: there is no $0'$ within $\text{CG}_\beta(t_\beta)$ matched to any 0 in the 0^{ℓ_4} in the beginning of the sequence. Suppose that there is such a pair. Then the $3\ell_2$ amount of $1'$ within $1'^{2\ell_2}$ in $\text{CG}_\alpha(t_\alpha)$ and within the first $1'^{\ell_2}$ in $\text{CG}_\beta(t_\beta)$ can only be matched to letters in $\text{CG}_\alpha(t_\alpha)$. However, the amount of 1 in $\text{CG}_\alpha(t_\alpha)$ is at most $2.1\ell_1$, so at least $0.9\ell_2$ amount of $1'$ are not matched. Therefore, $|A| \leq (6\ell_2 + 3.1\ell_1) - 0.9\ell_2 < \text{RNA}(0^{\ell_4}\text{CG}_\alpha(t_\alpha)0^{\ell_4}\text{CG}_\beta(t_\beta)0^{\ell_4}\text{CG}_\gamma(t_\gamma))$.
- Claim: there is no $0'$ within $\text{CG}_\alpha(t_\alpha)$ matched to any 0 in any 0^{ℓ_4} . Suppose that there is such a pair. We can show that at least ℓ_2 amount of $1'$ cannot participate in the RNA folding, so $|A| \leq (6\ell_2 + 3.1\ell_1) - \ell_2 < \text{RNA}(0^{\ell_4}\text{CG}_\alpha(t_\alpha)0^{\ell_4}\text{CG}_\beta(t_\beta)0^{\ell_4}\text{CG}_\gamma(t_\gamma))$.
 - Case: a $0'$ within $\text{CG}_\alpha(t_\alpha)$ is matched to a 0 in the first 0^{ℓ_4} . Then the $1'^{2\ell_2}$ in the beginning of $\text{CG}_\alpha(t_\alpha)$ cannot participate in the RNA folding.
 - Case: a $0'$ within $\text{CG}_\alpha(t_\alpha)$ is matched to a 0 in the second 0^{ℓ_4} . Then letters in the two $1'^{\ell_2}$ in $\text{CG}_\alpha(t_\alpha)$ can only be matched to letters within $p(\text{CLG}(t_\alpha)^R)$. Hence at least $2\ell_2 - 0.1\ell_1$ amount of 1 are unmatched. Since the difference between the number of 1 and $1'$ in $0^{\ell_4}\text{CG}_\alpha(t_\alpha)0^{\ell_4}\text{CG}_\beta(t_\beta)0^{\ell_4}\text{CG}_\gamma(t_\gamma)$ is at most $0.1\ell_1$, at least $2\ell_2 - 0.2\ell_1 > \ell_2$ amount of $1'$ cannot participate in the RNA folding.
 - Case: a $0'$ within $\text{CG}_\alpha(t_\alpha)$ is matched to a 0 in the third 0^{ℓ_4} . Then all $1'$ within $\text{CG}_\beta(t_\beta)$ can only be matched to 1 within $\text{CG}_\alpha(t_\alpha)$. It is obvious that the number of $1'$ within $\text{CG}_\beta(t_\beta)$ is at least ℓ_2 more than the number of 1 within $\text{CG}_\alpha(t_\alpha)$, so at least ℓ_2 amount of $1'$ cannot participate in the RNA folding.

Therefore,

$$\begin{aligned}
& \text{RNA}(0^{\ell_4}\text{CG}_\alpha(t_\alpha)0^{\ell_4}\text{CG}_\beta(t_\beta)0^{\ell_4}\text{CG}_\gamma(t_\gamma)) \\
&= \text{RNA}(\text{CG}_\alpha(t_\alpha)\text{CG}_\beta(t_\beta)\text{CG}_\gamma(t_\gamma)) \\
&= \text{RNA}(1'^{2\ell_2}p(\text{CLG}(t_\alpha)^R)0'^{\ell_1}1^{\ell_2}0^{\ell_1}\text{CNG}(t_\alpha)1^{\ell_2}1'^{\ell_2}p(\text{CLG}(t_\beta)^R)0'^{\ell_1}1'^{2\ell_2} \quad (\text{by definition}) \\
&\quad 0'^{\ell_1}p(\text{CNG}(t_\beta))1'^{\ell_2}1^{\ell_2}\text{CLG}(t_\gamma)^R0^{\ell_1}1^{\ell_2}0^{\ell_1}\text{CNG}(t_\gamma)1^{2\ell_2}) \\
&= \text{RNA}(1^{\ell_2}0^{\ell_1}\text{CNG}(t_\gamma)1^{2\ell_2}1'^{2\ell_2}p(\text{CLG}(t_\alpha)^R)0'^{\ell_1}1^{\ell_2}0^{\ell_1}\text{CNG}(t_\alpha)1^{\ell_2}1'^{\ell_2} \quad (\text{cyclic shift}) \\
&\quad p(\text{CLG}(t_\beta)^R)0'^{\ell_1}1'^{2\ell_2}0'^{\ell_1}p(\text{CNG}(t_\beta))1'^{\ell_2}1^{\ell_2}\text{CLG}(t_\gamma)^R0^{\ell_1}) \\
&= 4\ell_2 + \text{RNA}(1^{\ell_2}0^{\ell_1}\text{CNG}(t_\gamma)p(\text{CLG}(t_\alpha)^R)0'^{\ell_1}1^{\ell_2}0^{\ell_1}\text{CNG}(t_\alpha)p(\text{CLG}(t_\beta)^R)0'^{\ell_1} \quad (\text{Lemma 15}) \\
&\quad 1'^{2\ell_2}0'^{\ell_1}p(\text{CNG}(t_\beta))\text{CLG}(t_\gamma)^R0^{\ell_1}) \\
&= 4\ell_2 + \text{RNA}(S_{t_\alpha, t_\beta, t_\gamma}).
\end{aligned}$$

For the third equality, we just move $1^{\ell_2}0^{\ell_1}\text{CNG}(t_\gamma)1^{2\ell_2}$ from the end of the sequence to the beginning. The fourth equality follows by applying Lemma 15 iteratively (which removes $1^{2\ell_2}1'^{2\ell_2}$, $1^{\ell_2}1'^{\ell_2}$, and $1'^{\ell_2}1^{\ell_2}$). \square

By calculating the exact value of $\text{RNA}(S_{t_\alpha, t_\beta, t_\gamma})$, together with several previous lemmas, the next lemma shows that $\text{RNA}(S_G) = m_1 + m_2$.

Lemma 17. $\text{RNA}(S_G) = m_1 + m_2$.

Proof. In view of Lemma 5, 14, 16, it suffices to show that $\text{RNA}(S_{t_\alpha, t_\beta, t_\gamma}) = 2\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - \frac{1}{2}(\delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\beta)) + \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\gamma)) + \delta_{\text{LCS}}(\text{CLG}(t_\beta), \text{CNG}(t_\gamma)))$.

First of all, it is easy to observe that $\text{RNA}(S_{t_\alpha, t_\beta, t_\gamma}) \geq 2\ell_2 + 3\ell_1$, so for any optimal RNA folding A (of $S_{t_\alpha, t_\beta, t_\gamma}$), we must have $|A| \geq 2\ell_2 + 3\ell_1$.

We claim that in any optimal RNA folding A of $S_{t_\alpha, t_\beta, t_\gamma}$, the following two statements are true:

- For each of the two 1^{ℓ_2} , there is a 1 that is matched to a $1'$ in the $1'^{2\ell_2}$.
- For each of the $S_{t_\gamma, t_\alpha}, S_{t_\alpha, t_\beta}, S_{t_\beta, t_\gamma}$, there is a pair linking a $0'$ in its $0'^{\ell_1}$ and a 0 in its 0^{ℓ_1} .

For the first statement, suppose that one 1^{ℓ_2} does not have any letter matched to a $1'$ in the $1'^{2\ell_2}$. It is easy to observe that the number of $1'$ in $S_{t_\alpha, t_\beta, t_\gamma}$ that does not belong to $1'^{2\ell_2}$ is at most $0.1\ell_1$. Therefore, $|A|$ is at most the total number of $0'$ plus the total number of 1 minus $(\ell_2 - 0.1\ell_1)$. By a simple calculation, $|A| \leq 3.1\ell_1 + (2\ell_2 + 0.1\ell_1) - (\ell_2 - 0.1\ell_1) = \ell_2 + 3.3\ell_1 < \text{RNA}(S_{t_\alpha, t_\beta, t_\gamma})$. Therefore, we conclude the first statement.

For the second statement, suppose that there is an $S \in \{S_{t_\gamma, t_\alpha}, S_{t_\alpha, t_\beta}, S_{t_\beta, t_\gamma}\}$ that has no pair linking a $0'$ in its $0'^{\ell_1}$ and a 0 in its 0^{ℓ_1} . Due to the first statement, any pairing involving $0'^{\ell_1}$ and 0^{ℓ_1} are confined to be within S . Therefore, the number of pairs involving letters in S is at most $|S| - 2\ell_1 \leq 0.1\ell_1$. This is certainly not optimal, since simply matching all $0'$ in $0'^{\ell_1}$ to all 0 in 0^{ℓ_1} gives us ℓ_1 amount of pairs. Therefore, we conclude the second statement.

We can infer from the above two statements that for each $S \in \{S_{t_\gamma, t_\alpha}, S_{t_\alpha, t_\beta}, S_{t_\beta, t_\gamma}\}$, letters within S are only matched to letters within S in any optimal RNA folding of $S_{t_\alpha, t_\beta, t_\gamma}$.

As a result,

$$\begin{aligned}
\text{RNA}(S_{t_\alpha, t_\beta, t_\gamma}) &= \text{RNA}(1^{\ell_2} \circ 1^{\ell_2} \circ 1'^{2\ell_2}) + \text{RNA}(S_{t_\gamma, t_\alpha}) + \text{RNA}(S_{t_\alpha, t_\beta}) + \text{RNA}(S_{t_\beta, t_\gamma}) \\
&= 2\ell_2 + 3\ell_1 + \text{RNA}(\text{CNG}(t_\gamma)p(\text{CLG}(t_\alpha)^R)) + \text{RNA}(\text{CNG}(t_\alpha)p(\text{CLG}(t_\beta)^R)) \\
&\quad + \text{RNA}(p(\text{CNG}(t_\beta))\text{CLG}(t_\gamma)^R) \\
&= 2\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - \frac{1}{2}(\delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\beta)) + \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\gamma)) \\
&\quad + \delta_{\text{LCS}}(\text{CLG}(t_\beta), \text{CNG}(t_\gamma))).
\end{aligned}$$

□

We are ready to prove the main theorem of the paper:

Remainder of Theorem 2. *If the RNA folding problem on sequences in $\{A, C, G, U\}^n$ can be solved in $T(n)$ time, then $3k$ -clique on graphs with $|V| = n$ can be solved in $\mathcal{O}(T(n^{k+1} \log(n)))$ time.*

Proof. Given a graph G , we construct the string S_G . According to Lemma 1, 4, the length of S_G is $\mathcal{O}(k^2 n^{k+1} \log(n))$, and S_G can be constructed in time $\mathcal{O}(k^2 n^{k+1} \log(n))$.

We let $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$ be chosen such that $Q = \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\beta)) + \delta_{\text{LCS}}(\text{CLG}(t_\alpha), \text{CNG}(t_\gamma)) + \delta_{\text{LCS}}(\text{CLG}(t_\beta), \text{CNG}(t_\gamma))$ is minimized. By Lemma 3, there exists a number c_1 such that:

- the number c_1 depends only on n, k , and $Q \geq 3c_1$.
- If $Q = 3c_1$, then each of $t_\alpha \cup t_\beta$, $t_\alpha \cup t_\gamma$, $t_\beta \cup t_\gamma$ is a $2k$ -clique, which in turn is equivalent to “ $t_\alpha \cup t_\beta \cup t_\gamma$ is a $3k$ -clique”.
- If $Q > 3c_1$, then the graph has no $3k$ -clique.

According to Lemma 17, $\text{RNA}(S_G) = m_1 + m_2$. By its definition, m_1 only depends on n, k , and $m_2 = 6\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - \frac{Q}{2}$. Hence we are able to decide whether G has a $3k$ -clique from the value of $\text{RNA}(S_G)$, which can be calculated in time $T(\mathcal{O}(k^2 n^{k+1} \log(n))) = \mathcal{O}(T(k^2 n^{k+1} \log(n)))$.

Note that k is treated as a constant instead of an input parameter. □

4. Hardness of Dyck Edit Distance Problem

In this section, we shift our focus to the Dyck edit distance problem. We will present a simple reduction from RNA folding problem (with alphabet size 4) to Dyck edit distance problem (with alphabet size 10). This leads to a much simplified and improved proof for a conditional lower bound of Dyck edit distance based on the conjectured hardness k -clique (the previous proof presented in [3] requires 48 symbols).

Dyck Edit Distance. Given $S \in (\Sigma \cup \Sigma')^n$, the goal of the Dyck edit distance problem is to find a minimum number of edit operations (insertion, deletion, and substitution) that transform S into a string in the Dyck context free language.

Given Σ and its corresponding Σ' , the Dyck context free language is defined by the grammar with following production rules: $\mathbf{S} \rightarrow \mathbf{SS}$, $\forall x \in \Sigma, \mathbf{S} \rightarrow x\mathbf{S}x'$, and $\mathbf{S} \rightarrow \epsilon$ (empty string).

An alternative definition of the Dyck edit distance problem is described as follows:

Given a sequence $S \in (\Sigma \cup \Sigma')^n$, find a minimum cost set $A \subseteq \{(i, j) | 1 \leq i < j \leq n\}$ satisfying the following conditions:

- $A = A_M \uplus A_S$ has no crossing pair.
- A_M contains only pairs of the form (x, x') , $x \in \Sigma$ (i.e. for all $(i, j) \in A_M$, we have $S[i] = x$, $S[j] = x'$, for some $x \in \Sigma$). A_M corresponds to the set of matched pairs.
- A_S does not contain any pair of the form (y', x) , $x, y \in \Sigma$ (i.e. for all $(i, j) \in A_S$ we have either $S[i] \in \Sigma$ or $S[j] \in \Sigma'$). A_S corresponds to the set of pairs that can be fixed by one substitution operation per each pair.
- Let D be the set of letters in S that do not belong to any pair in A . Each letter in D requires one deletion/insertion operation to fix.

The cost of A is then defined as $|A_S| + |D|$, and the Dyck edit distance of the string S is the cost of a minimum cost set meeting the above conditions.

Dyck edit distance problem can be thought of as an asymmetric version of the RNA folding problem (in RNA folding, we allowed the aligned pair to be either (x, x') or (x', x) , $x \in \Sigma$) that also handles substitution (in addition to deletion and insertion). Though these two problems look similar, they can behave quite differently. For example, in Section 1 we describe a simple reduction from LCS to RNA folding; since LCS is edit distance problem without substitution, one may hope that the same reduction reduces edit distance problem to Dyck edit distance problem. However, this is not true due to the following counterexample: both the two strings $ababa$ and $abbaa$ require at least 4 edit operations to transform into the string $caaac$; but the Dyck edit distance of $ababac'a'a'a'c'$ is 4 (by deleting all b, c'), while the Dyck edit distance of $abbaac'a'a'a'c'$ is 3 (by deleting all c' and substituting the second b with b').

Intuitively, the substitution operation makes Dyck edit distance more complicated than RNA folding. Indeed, the same conditional lower bound as Theorem 1 for Dyck edit distance problem shown in [3] requires a bigger alphabet size (48 instead of 36) and a longer proof.

In the next, we prove Theorem 3 by demonstrating a simple reduction from RNA folding problem to Dyck edit distance problem with alphabet size 10. This improves upon the hardness result in [3], and justifies the intuition that Dyck edit distance is a harder problem than RNA folding.

Proof of Theorem 3. *If Dyck edit distance problem on sequences of length n with alphabet size 10 can be solved in $T(n)$ time, then the RNA folding problem on sequences in $\{A, C, G, U\}^n$ can be solved in $\mathcal{O}(T(n))$ time.*

Proof. For notational simplicity, we let the alphabet for the RNA folding problem be $\Sigma \cup \Sigma' = \{0, 0', 1, 1'\}$ (instead of $\{A, C, G, U\}$). Let S be any string in $(\Sigma \cup \Sigma')^n$. We define the string S_{Dyck} as the result of applying the following operations on S :

- Replace each letter 0 with the sequence $S_0 = aeb' aeb'$.
- Replace each letter $0'$ with the sequence $S_{0'} = bba' a'$.
- Replace each letter 1 with the sequence $S_1 = ced' ced'$.
- Replace each letter $1'$ with the sequence $S_{1'} = ddc' c'$.

It is clear that S_{Dyck} is a sequence of length at most $6n$ on the alphabet $\{a, b, c, d, e\} \cup \{a', b', c', d', e'\}$, though the letter e' is not used. We claim that the Dyck edit distance of S_{Dyck} is $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$.

First, we show that the Dyck edit distance of S_{Dyck} is at most $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$. Given an optimal RNA folding of S , we construct a crossing-free matching A with cost $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$ as follows:

For matched pairs in the RNA folding of S :

- For each matched pair $(0, 0')$ in the RNA folding of S , we add two pairs $(a, a'), (a, a')$ to A_M , and add three pairs $(e, b'), (e, b'), (b, b)$ to A_S in its corresponding pair of substrings $(S_0 = \mathbf{a}(eb')\mathbf{a}(eb'), S_{0'} = (bb)\mathbf{a}'\mathbf{a}')$ in S_{Dyck} .
- For each matched pair $(0', 0)$ in the RNA folding of S , we add two pairs $(b, b'), (b, b')$ to A_M , and add three pairs $(a', a'), (a, e), (a, e)$ to A_S in its corresponding pair of substrings $(S_{0'} = \mathbf{bb}(a'a'), S_0 = (ae)\mathbf{b}'(ae)\mathbf{b}')$ in S_{Dyck} .
- Similarly, for each matched pair $(1, 1'), (1', 1)$ in the RNA folding of S , we can add two pairs to A_M and three pairs to A_S .

For unmatched letters in S :

- For each unmatched letter 0 in S , we add three pairs $(a, b'), (e, b'), (a, e)$ to A_S in its corresponding substring $S_0 = (a(eb')(ae)b')$. Similarly, for each unmatched letter 1, we can add three pairs to A_S .
- For each unmatched letter $0'$ in S , we add two pairs $(b, b), (a', a')$ to A_S in its corresponding substring $S_{0'} = (bb)(a'a')$. Similarly, for each unmatched letter $1'$, we can add two pairs to A_S .

The set A_M has size $2\text{RNA}(S)$, the set A_S has size $\frac{|S_{\text{Dyck}}| - 4\text{RNA}(S)}{2}$, and D is an empty set. Therefore, the cost of A is $\frac{|S_{\text{Dyck}}| - 4\text{RNA}(S)}{2} = \frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$.

Second, we show that the Dyck edit distance of S_{Dyck} is at least $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$. Given a crossing-free matching A (on the string S_{Dyck}) of cost C , we recover an RNA folding of S that has $\geq \frac{|S_{\text{Dyck}}|}{4} - \frac{C}{2}$ number of matched pairs.

We build a multi-graph $G = (V, E)$ such that V is the set of all substrings $S_0, S_{0'}, S_1, S_{1'}$ that constitute S_{Dyck} , and the number of edges between two substrings in V is the number of pairs in A_M linking letters between these two substrings. Note that $|V| = n, |E| = A_M$. It is clear that $C \geq \frac{|S_{\text{Dyck}}| - 2|E|}{2}$ (since $|A_S| + |D| \geq \frac{|S_{\text{Dyck}}| - 2|A_M|}{2} = \frac{|S_{\text{Dyck}}| - 2|E|}{2}$). Therefore, we are done if we can recover an RNA folding of size $\geq \frac{|E|}{2}$, since $\frac{|E|}{2} \geq \frac{|S_{\text{Dyck}}|}{4} - \frac{C}{2}$.

We observe the following:

- G has degree at most 2 (due to our definition of $S_0, S_{0'}, S_1, S_{1'}$, at most two letters in such a substring can participate in pairings of the form (x, x') , $x \in \{a, b, c, d\}$, without crossing).
- In the graph G , any edge must either links an S_0 with an $S_{0'}$ or links an S_1 with an $S_{1'}$ (due to our definition of $S_0, S_{0'}, S_1, S_{1'}$, any pairing of the form (x, x') , $x \in \{a, b, c, d\}$, must be made between $S_0, S_{0'}$ or between $S_1, S_{1'}$).
- G does not contain any cycle of odd length (due to the above observation).

In view of the above (second) observation, a (graph-theoretic) matching $M \subseteq E$ of G naturally corresponds to a (size $|M|$) RNA folding of S : for each edge (a pair of substrings in S_{Dyck}) in M , we add its corresponding pair of letters in S to the RNA folding. Since a maximum matching has size $\geq \frac{|E|}{2}$ in a graph of maximum degree 2 without odd cycles, we conclude the proof. \square

We note that for the case substitution is not allowed, the letter e in the above proof is not needed, and this lowers the required alphabet size to 8.

The reason that the letter e is essential for the above proof to work is explained as follows: Suppose that e is removed. For each matched pair $(0, 0')$ in the RNA folding of S , after adding two pairs $(a, a'), (a, a')$ to A_M , the letter b' between two a in $S_0 = ab'ab'$ cannot participate in any matching anymore. Hence some letters will be in D according to our construction of the crossing-free matching A . This indicates that our construction may not be optimal. Indeed, for the string $(00'0')_{\text{Dyck}} = ab'ab'bba'a'bba'a'$ (after removing e), if we insist on matching the two pairs $(a, a'), (a, a')$ in $\mathbf{a}b'\mathbf{a}b'bba'\mathbf{a'bba'a'}$, then the cost will be at least 5 (three substitutions and two deletions are needed). However, there is a solution that uses only 4 substitutions: $\mathbf{a}(b'\mathbf{a}(b'(bb)a')\mathbf{a'}(bb)a')\mathbf{a'}$.

5. Conclusion and Future Directions

In this paper we present a hardness result of RNA folding problem with alphabet size 4 and demonstrate a reduction from RNA folding problem to Dyck edit distance problem. A few open problems still remain:

- There are still a few cases where the state-of-art conditional lower bound requires a certain alphabet size to work (e.g. Theorem 3, Corollary 1, and the hardness result for Dynamic time warping in [8]). Is it possible to improve them using our technique or other ideas?
- Is it possible to reduce Dyck edit distance problem to RNA folding problem?
- Besides the classic RNA folding problem, several problems in bioinformatics admit similar formulation (see e.g. [2,13]). It would be interesting to see whether the technique presented in this paper (and [3,8]) can be adapted to give meaningful lower bounds for other problems.

Acknowledgements. The author would like to thank Seth Pettie for helpful discussions and comments.

References

1. Tatsuya Akutsu. Approximation and exact algorithms for RNA secondary structure prediction and recognition of stochastic context-free languages. *Journal of Combinatorial Optimization*, 3(2): 321-336, 1999.
2. Mika Amit, Rolf Backofen, Steffen Heyne, Gad M. Landau, Mathias Mohl, Christina Schmiedl, Sebastian Will. Local Exact Pattern Matching for Non-fixed RNA Structures. In *Combinatorial Pattern Matching (CPM)*, 306–320, Springer, 2012.

3. Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant’s Parser. In *Proceedings of IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, 2015.
4. Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Quadratic-time hardness of LCS and other sequence similarity measures. In *Proceedings of IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, 2015.
5. Amihood Amir, Timothy M. Chan, Moshe Lewenstein, and Noa Lewenstein. On hardness of jumbled indexing. In *Proceedings of the 41st International Colloquium Automata, Languages, and Programming (ICALP)*, 114–125, 2014.
6. Amihood Amir and Gad M. Landau. Fast parallel and serial multidimensional approximate array matching. *Theoretical Computer Science*, 81(1): 97–115, 1991.
7. Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 51–58, 2015.
8. Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proceedings of IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, 79–97, 2015.
9. Timothy M. Chan and Moshe Lewenstein. Clustered integer 3SUM via additive combinatorics. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 31–40, 2015.
10. Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme J. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
11. Friedrich Eisenbrand and Fabrizio Grandoni. On the complexity of fixed parameter clique and dominating set. *Theoretical Computer Science*, 326(1): 57–67, 2004.
12. Yelena Frid and Dan Gusfield. A simple, practical and complete $\mathcal{O}(\frac{n^3}{\log n})$ -time algorithm for RNA folding using the four-russians speedup. *Algorithms for Molecular Biology*, 5(1):13, 2010.
13. Yelena Frid, Dan Gusfield: Speedup of RNA Pseudoknotted Secondary Structure Recurrence Computation with the Four-Russians Method. In *Combinatorial Optimization and Applications (COCOA)*, 176–187, 2012.
14. Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1065–1075, 2010.
15. Tamar Pinhas, Dekel Tsur, Shay Zakov, and Michal Ziv-Ukelson. Edit distance with duplications and contractions revisited. In *Combinatorial Pattern Matching (CPM)*, 441–454, Springer, 2011.
16. Tamar Pinhas, Shay Zakov, Dekel Tsur, and Michal Ziv-Ukelson. Efficient edit distance with duplications and contractions. *Algorithms for Molecular Biology*, 8(1):27, 2013.
17. Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, 515–524, 2013.
18. Barna Saha. The dyck language edit distance problem in near-linear time. In *Proceedings of the IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, 611–620, 2014.
19. Barna Saha. Language edit distance & maximum likelihood parsing of stochastic grammars: faster Algorithms & connection to fundamental graph problems. In *Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, 118–135, 2015.
20. Yinglei Song. Time and space efficient algorithms for RNA folding with the four-russians technique. *Electronic preprint arXiv:1503.05670*, 2015.
21. Balaji Venkatachalam, Dan Gusfield, and Yelena Frid. Faster algorithms for RNA-folding using the four-russians method. In *Algorithms in Bioinformatics (WABI)*, 126–140, Springer, 2013.
22. Leslie G. Valiant. General context-free recognition in less than cubic time. *Journal of Computer and System Sciences*, 10(2): 308–315, 1975.
23. Virginia Vassilevska. Efficient algorithms for clique problems. *Information Processing Letters*, 109(4): 254–257, 2009.